NIST
National Institute of
Standards and Technology
Technology Administration
U.S. Department of Commerce

Special Publication 800-81r1
Unofficial Errata

Version 1.0
9/4/2010

This is an unofficial update/changes to NIST Special Publication 800-81r1. The Special Publication was released while the Internet community is still in the process on developing best common practices and improving the security of DNSSEC. These changes to not alter the main checklist items in the guide only refine the text to keep the document up to date with respects to ongoing work from the Internet community.

This document will be updated on an irregular basis as new material is produced, best common practices are refined, and operators gain more experience with DNSSEC. This document will remain "unofficial" in that it is not part of the official Special Publication errata process for now. An official version may be published in the future.

In the text below, the text that appears in the NIST SP 80-81r1 guide is marked in blue, changes in black, and any text that appears in *italics* describes the reasoning behind the change or addition.

6.1.4 Protection Approach for DNS Query/Response Threats—DNSSEC

The underlying feature in the major threat associated with DNS query/response (i.e., forged response or response failure) is the integrity of DNS data returned in the response. Hence, the security objective is to verify the integrity of each response received. An integral part of integrity verification is to ensure that valid data has originated from the right source. Establishing trust in the source is called *data origin authentication*. Hence, the security objectives—and consequently the security services—that are required for securing the DNS query/response transaction are data origin authentication and data integrity verification.

These services could be provided by establishing trust in the source and verifying the signature of the data sent by that source. The specification for a digital signature mechanism in the context of the DNS infrastructure is in IETF's DNSSEC standard. The objectives, additional RRs, and DNS message contents involved in the DNSSEC are specified through RFCs 4033, 4034, and 4035 [RFC4033], [RFC4034], [RFC4035]. In DNSSEC, trust in the public key (for signature verification) of the source is established not by going to a third party or a chain of third parties (as in public key infrastructure [PKI] chaining), but by starting from a trusted name server (such as the root name server) and establishing the chain of trust down to the current source of response through successive verifications of signature of the public key of a child by its parent. The public key of the trusted name servers is called the *trust anchor*.

After authenticating the source, the next process DNSSEC calls for is to authenticate the response. This requires that responses consist of not only the requested RRs but also an authenticator associated with them. In DNSSEC, this authenticator is the digital signature of an RRSet. The digital signature of an RRSet is encapsulated through a special RRType called RRSIG. The DNS client using the trusted public key of the source (whose trust has just been established) then verifies the digital signature to detect if the response is valid or bogus.

To ensure that RRs associated with a query are really missing in the zone file and have not been removed in transit, the DNSSEC mechanism provides a means for authenticating the nonexistence of an RR. It generates a special RR called an NSEC RR (or NSEC3 RR) that lists the RRTypes associated with an owner name as well as the next name in the zone file. It sends this special RR, along with its signature, to the resolving name server. By verifying this signature, a DNSSEC-aware resolving name server can determine which authoritative owner name exists in a zone and which authoritative RRTypes exist at those owner names.

To protect against the threat of incorrect application of expansion rules for wildcard RRs, the DNSSEC mechanism provides a means of comparing the validated wildcard RR against an NSEC RR (or NSEC3 RR) and thereby verifying that the name server applied the wildcard expansion rules correctly in generating an answer.

DNSSEC can guarantee the integrity of name resolution responses to DNS clients acting on behalf of Internet-based resources, provided the clients perform the DNSSEC signature verification. In many cases, however, these DNS clients are stub resolvers that are not DNSSEC-aware. If signature verification is performed by the resolving name server providing name resolution service for the clients that are stub resolvers, the end-to-end integrity of the response data can be guaranteed only by protecting the communication channel between the resolving name server and the stub resolver. IETF’s design criteria consider DNS data to be public; hence, confidentiality is not one of the security goals of DNSSEC. DNSSEC is not designed to directly protect against denial-of-service threats, although it does so indirectly by providing message integrity and source authentication. DNSSEC also does not provide communication channel security because name resolution queries and responses travel over millions of nodes of the public Internet. DNSSEC also can lead to a new type of weakness that did not exist in DNS before. An artifact of how DNSSEC peforms negative responses allows a client to map all the names in a zone. This is called Zone Walking. Zone Walking provides an attacker with a “map” of a target zone with all domain names and IP addresses in the zone and enables him/her to determine the configuration of the internal network and launch some targeted attacks on some key hosts. Therefore, it is advisable that a zone only contains zone data that the administrator wants to be made public or use the NSEC3 RR option for providing authenticated denial of existence. For internal DNS, something like split-DNS (see Section 7.2.8) could be deployed. For NSEC3, see Section 10.4.

Reason for change: *NSEC3 is a variant for DNSSEC was not stated in the original SP 800-81r1, and was actually developed after the initial release. NSEC3 provides the same security features as NSEC, but uses hashed owner names to increase the work an attacker needs to do to conduct a Zone Walking attack.*

6.2 Zone Transfer Threats and Protection Approaches

Zone transfers are performed to replicate zone files in multiple servers to provide a degree of fault tolerance in the DNS service provided by an organization. Threats from zone transfers have not been documented formally through any IETF RFCs. A few threats could be expected, however: the first threat, denial of service, is common for any network transaction. The second threat is based on exploitation of knowledge gained from the information provided by zone transfers. The third threat is common to any network packet.

- **Threat T15—Denial of Service:** Because zone transfers involve the transfer of entire zones, they place substantial demands on network resources relative to normal DNS queries. Errant or malicious frequent zone transfer requests on the name servers of the enterprise can overload the master zone server and result in denial of service to legitimate users.

- **Threat T16—**The zone transfer response message could be tampered.

The denial-of-service can be minimized if servers allowed to make zone transfer requests are restricted to a set of known entities. To configure this restriction into the primary name server, there should be a means of identifying those entities. Name server software such as BIND initially provided a configuration feature to restrict zone transfer requests to a set of designated IP addresses. Because IP addresses can be spoofed, however, this mode of configuration does not provide an adequate means of restricting zone transfer access.

The IETF developed an alternate mechanism called a *transaction signature* (TSIG), whereby mutual identification of servers is based on a shared secret key. Because the number of servers involved in zone transfer is limited (generally restricted to name servers in the same administrative domain of an organization), a bilateral trust model that is based on a shared secret key may be adequate for most enterprise (except for very large ones). TSIG specifies that the shared secret key be used not only for mutual authentication but also for signing zone transfer requests and responses. Hence, it provides protection against tampering of zone transfer response messages (threat T15). Protection of DNS data alone (the payload) in a zone transfer message also can be ensured through verification of signature records accompanying RRs from a DNSSEC-signed zone. These signatures, however, do not cover all the information in a zone file (e.g., delegation information). Furthermore, they enable verification of only the individual RRsets and not the entire zone transfer response message.

There is also another method to authenticate DNS transactions by using asymmetric cryptography (i.e. public key cryptography). The format of the SIG(0) RR is similar to the resource record signature (RRSIG) RR (see Section 9.2.1), and can be validated using a public key stored in the DNS (instead of a shared secret key). SIG(0) can be more computational expensive to use, but offer an advantage in that a previous trust relationship may not be necessary to use SIG(0) signed messages. However, since most zone transfers occur between parties that have a previously established relationship, it is considered easier to implement TSIG for authenticating zone transfer transactions.

Another possibility is to rely on lower level network layer to provide security such as IPSec. This would remove the need for authentication at the DNS (application) layer. How to set up this level of security is beyond the scope of this guide.

Reason for change: *IPSec is a valid alternative to provide protection for DNS zone transfer transactions. IPSec provides authentication the lower network layer so there would be no need for TSIG or SIG(0) authentication at the application (i.e. DNS) layer.*

8.2.5 Checklists for Key File Creation and Key Configuration Process

Checklist item 8: The TSIG key should be a minimum of 112 bits in length if the generator utility has been proven to generate sufficiently random strings [800-57P1]. The generated TSIG key may have to be longer to insure at least 112 bits of security (128 bits recommended).

Reason for the change: *Most random number generation software packages cannot guarantee complete randomness, so 128 bit TSIG shared secret strings are recommended (at a minimum). Longer values are also acceptable, depending on the local policy.*

8.3 Recommendations Summary

- **Checklist item 8:** The TSIG key should be a minimum of 112 bits in length (128 bit recommended) if the generator utility has been proven to generate sufficiently random strings [800-57P1].

Reason for the change: *To keep Checklist item in line with change to Section 8.2.5 above.*

9.3 Generation of Public Key-Private Key Pair (DNSSEC-OP1)

DNSSEC specifies generation and verification of digital signatures using asymmetric keys. This requires generation of a public key-private key pair. Although the DNSSEC specification does not call for different keys (just one key pair), experience from pilot implementations suggests that for easier routine security administration operations such as key rollover (changing of keys) and zone re-signing, at least two different types of keys are needed. One set is called Key Signing Key (KSK). This key (specifically, the private part of the key pair, called KSK-private) will be used only for signing the key set (i.e., DNSKEY RRSet) in the zone file. The other key type is called the Zone Signing Key (ZSK) (whose private part is called ZSK-private) and will be used to sign all RRsets in the zone (including DNSKEY RRSet). An administrative distinction is made between the KSK and ZSK keys by setting the Secure Entry Point (SEP) flag bit in the DNSKEY RR that represents the public part of those keys (in this case, it would be called KSK-public).

The logic behind creation of two types of key pairs is to provide separate set of functions for each key type and thus reduce the overall complexity of tasks involved in key rollovers and zone re-signing. Accordingly, the KSK (KSK-private) is used to sign the key set (i.e., DNSKEY RRSet) and is the key type (public component – KSK-public) that is sent to the parent to be used for authenticated delegation. This is done by generating a DS RR, using the hash of the child's KSK-public key and generating a corresponding signature (RRSIG RR) using the parent's own ZSK. The KSK (KSK-public) may also be used as a trust anchor (sometimes called the SEP keys) in validating resolvers to establish trust chains for verification of signatures.

The ZSK (ZSK-private) is to be used for signing the entire zone file (all RRsets). The public portion of this key (ZSK-public) will not be sent to the parent and will always remain in the zone.
The decision parameters involved in KSK and ZSK key pair generation are as follows:

- Choice of digital signature algorithm
- Choice of key sizes
- Choice of crypto period (duration for which the key will be used).

The choice of digital signature algorithm will be based on recommended algorithms in well known standards. NIST's Digital Signature Standard (DSS) [FIPS186] provides three algorithm choices:

- Digital Signature Algorithm (DSA)
- RSA
- Elliptic Curve DSA (ECDSA).

Of these three algorithms, RSA and DSA are more widely available and hence are considered candidates of choice for DNSSEC. In terms of performance, both RSA and DSA have comparable signature generation speeds, but DSA is much slower for signature verification. Hence, RSA is the recommended algorithm as far as this guideline is concerned. RSA with SHA-1 is currently the only cryptographic algorithm mandated to be implemented with DNSSEC although other algorithm suites (i.e. RSA/SHA- 256) are also specified. It can be expected that name servers and clients will be able to use the RSA algorithm at the minimum. It is suggested that at least one ZSK for a zone use the RSA algorithm.

NIST's Secure Hash Standard (SHS) (FIPS 180-3) specifies SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512 as approved hash algorithms to be used as part of the algorithm suite for generating digital signatures using the digital signature algorithms in the NIST's DSS[FIPS186]. It is expected that there will be support for Elliptic Curve Cryptography in the DNSSEC. The migration path for USG DNSSEC operation will be to ECDSA (or similar) from RSA/SHA-1 and RSA/SHA-256 before September 30th, 2015.

The choice of key size is a tradeoff between the risk of key compromise and performance. The performance variables are signature generation and verification times. The size of the DNS response packet also is a factor because DNSKEY RRs may be sent in the additional section of the DNS response. Because the KSK is used only for signing the key set (DNSKEY RRSet), performance is not much of an issue. Compromise of a KSK could have a great impact, however, because the KSK is the entry point key for a zone. Rollover of a KSK in the event of a compromise involves potential update of trust anchors in many validating resolvers.

As far as the choice of key size for the ZSK is concerned, performance certainly will be a factor because the ZSK is used for signing all RRsets in the zone. In terms of impact, however, it is restricted to just a single zone because the ZSK's usage is limited to signing RRsets only for that zone. This is the justification for allowing 1024 bit RSA keys for use with DNSSEC beyond the USG stop date of 2010. Some network components have been shown to have problems handling large DNS responses. The use of 1024 bit RSA keys is still considered acceptable to compensate for this as long as other rigorous key management practices are in place.

The choice of crypto period (rollover period) is dictated by the amount of work required to compromise the given key. The large size of the KSK implies that the crypto period for that key can be long (usually a year or two). This aids in DNS operations as well, as KSK rollover is more disruptive and requires the zone administrator to interact with their parent zone to update the KSK's DS RR in the parent delegation information. In the case of ZSK, the risk of key guessing is higher of its smaller size. This implies that ZSKs must be rolled over more frequently than KSKs (usually between 1-3 months). Since the ZSK is local to the zone, rolling the ZSK is not as disruptive as rolling the KSK.

In the case of ZSK, the risk of compromise is greater due to the more frequent use. If the zone allows dynamic update, the ZSK is often stored on the same server as the zone. This factor, combined with the relatively smaller size of the key, implies that ZSKs must be rolled over more frequently than KSKs (usually between 1-3 months).

In terms of the number of keys of each type (KSK and ZSK) to be generated, a good practice is to generate an extra ZSK in addition to the one that will be used for signing. Hence, the zone administrator should use the key generation program to generate one KSK and two ZSKs during initial deployment of DNSSEC. One ZSK is treated as the active key, and its private part (ZSK-private) will be used for signature generation. The other ZSK (ZSK-public) will be made part of DNSKEY RRSet, but its associated private part (ZSK-private) will not be used for signing RRsets. This additional ZSK will provide a readily available ZSK for immediate rollover in emergency situations such as key compromise and a form of advance notification to validating resolvers that this key is to be the one into which the zone is going to roll over after the current crypto period expires. The mere presence of the key in the DNSKEY RRSet enables validating resolvers to cache and establish trust in the new key so that they can immediately use the key for signature verification as soon as rollover occurs.
The recommended digital signature algorithm suite, key sizes, and crypto periods for the KSK and

ZSK keys are given in Table 9-1 [800-57P1]. As with all data authentication keys, this table assumes approved components7 (hardware or software) and management operations are in place within the organization.

Key Type	Digital Signature Algorithm Suite	Key Size	Crypto Period (Rollover Period)
Key-Signing Key (KSK)	RSA-SHA1 (RSA-SHA-256) until 2015	2048 bits	12-24 months (1-2 years)
Zone-Signing Key (ZSK)	RSA-SHA1 (RSA-SHA-256) until 2015	1024 bits	1-3 months (30-90 days)

Table 9-1. Digital Signature Algorithms, Min. Key Sizes, and Crypto Periods

In the above table, the digital signature algorithm suite is given as both RSA-SHA1 and RSA-SHA256. This is because as of the time of writing, RSA-SHA1 is the only algorithm that is both Mandatory for implementations and Approved for use in the Federal Government. However, RSA-SHA1 will be phased out and replaced by RSA-SHA256 within the Federal Government. It is expected that not all software will be updated – especially outside the Federal Government. Because of this, DNS administrators may wish to deploy and use both algorithms for a period of time so DNSSEC client software that does not understand RSA-SHA256 can still get some protection from DNSSEC. The DNS root zone uses RSA/SHA-256 for signing, so deployment of RSA/SHA-256 enabled DNS validators has quickened. It is recommended that new DNSSEC deployments consider using RSA/SHA-256, rather than going through the complicated process of algorithm rollover.

The use of RSA in DNSSEC is approved until the year 2015. By this time, it is expected that Elliptic Curve Cryptography will be specified in the DNSSEC. USG DNS administrators should plan to migrate to the use of ECDSA (or similar) when it becomes available in DNSSEC components.

Reason for the change: *Text changed for correctness (key lifetime, cryptoanalysis, etc.). Added text about DNSSEC at the root zone and how it will speed deployment of RSA/SHA-256 in validating resolvers. Therefore, it is recommended for new DNSSEC deployments to start with RSA/SHA-256 rather than RSA/SHA-1 and then go through the complicated process of algorithm rollover. Since ECDSA is not available in DNS software, it is not possible to initially sign using ECDSA.*

11.2 Scheduled Key Rollovers (Key Lifetimes)

The keys used for zone signing (ZSK) and key signing (KSK) have to be changed because the keys become vulnerable (liable to be cracked) after a period of usage (generally attributed to Moore's Law—which predicts an exponential increase in computing power over time—but also because of other factors discussed below). The compromise of a private key means that any site can spoof the zone by signing a bogus RRSet with the private key, thus defeating the purpose of signing the zone file. Key rollover can take place as a scheduled event (scheduled rollover), or it may take place as a result of an emergency (emergency rollover). Emergency rollover occurs for one of the following reasons:

- The private key of the zone has been compromised.
- The private key of the zone has been lost, and the zone is to be updated before the RRSIGs expire.

In scheduled key rollover, the time period (or frequency of change) after which the keys must be changed is determined by several factors.

- The amount of effort needed to rollover the key, and the potential disruption to the zone or current operation.

- Information security policy within the organization.

- The smaller the size of the private key, the easier it is to crack.

Based on these factors, each zone arrives at a desired frequency for key rollovers for ZSKs and KSK. Recall that the KSK (KSK-private) is used for signing only the DNSKEY RRSet, whereas the ZSK (ZSK-private) is used for signing the entire zone file. Apart from the volume of data, the ZSK also is used much more frequently, as in the following situations:

- When a new RR is added (e.g., a new mail server is added and hence a new MX RR is added the zone file)

- When an existing RR's RDATA has changed (e.g., the IP address of a server has changed and hence the corresponding A RR has to be replaced)

- When the signature has expired for an RRSIG RR.

Because of the volume of data handled and the frequency of usage, the size of the ZSK-private key becomes a factor in overall CPU cycles consumed by the digital signature generation operations. Hence, the ZSK used is often relatively small.

Checklist item 28: The KSK needs to be rolled over less frequently than the ZSK. The recommended rollover frequency for the KSK is once every 1-2 years, whereas the ZSK should be rolled over every 1-3 months for operational consistency but may be used longer if necessary for stability. Both keys should have an Approved length according to NIST SP 800-57 Part 1 [800-57P1], [800-57P3].

The impact of a key rollover on the rest of DNS depends on whether the secure zone is locally secure or globally secure (part of a chain of trust).

For a more detailed discussion of the operational steps involved in a key rollover, see the IETF document on DNSSEC operations [RFC4641]. Note that the two processes described for key rollovers (pre-published and dual-signature) can be used in rolling over the ZSK or the KSK. The recommendations below are based on common practice and minimizing the impact of larger responses on clients.

Reason for Change: *Text on reasoning changed for correctness and to bring section in line with previous text on key length recommendations.*

11.2.1 Key Rollover in a Locally Secure Zone

A zone that is only locally secure will have a ZSK, and possibly a KSK that is configured in client resolvers, as a trusted key. Certain challenges arise when either key is rolled over, although having a KSK even for a locally signed zone makes rolling over the ZSK easier. When a zone changes its ZSK(s) and has a KSK that remains unchanged, the only problem that must be addressed is introducing the new key when the old key may be in some distant resolver's or name server's cache. The solution is to pre-publish the new public key before the rollover. The DNS administrator needs to publish the new key as a DNSKEY RR in the zone file before it is used to generate signatures. The process is as follows:

- Generate a new key pair.
- Add the public key of the new key pair to the zone file (DNSKEY record).
- Sign the zone using the private key of the currently active key pair and the KSK (if present).
- Wait for a period equal to the TTL of the DNSKEY RRSet or the MinTTL of the zone SOA record (whichever is greater).
- Delete the RRSIG RRs generated by the outgoing key, but retain the DNSKEY RR. Resign the zone using the new ZSK (and current KSK, if used).
- Wait the TTL of the zone's DNSKEY RRset
- Remove the old, outgoing ZSK from the DNSKEY RRset.
- Re-sign the DNSKEY RRSet with the new ZSK.

It might be in the DNS administrator's best interest to perform a ZSK rollover continuously. The administrator can perform the first three steps and wait indefinitely before deleting the old DNSKEY from the key set, even continuing to sign the zone with the old DNSKEY when the RRSIGs in the zone expire. This procedure allows the administrator to perform an emergency key rollover more efficiently (see below).

Zones that pre-publish the new public key should observe the following:

Checklist item 29: The secure zone that pre-publishes its public key should do so at least one TTL period before the start of the key rollover.

Checklist item 30: After removing the old public key, the zone should generate a new signature (RRSIG RR), based on the remaining keys (DNSKEY RRs) in the zone file.

In rolling over the KSK, the secure zone may not know which resolvers have stored the public key as a trust anchor. If the network administrator has an out-of-band method of contacting resolver administrators that have stored the public key as a trust anchor (such as e-mail), the network administrator should send out appropriate warnings and set up a trusted means of disseminating the new trust anchor. Otherwise, the DNS administrator can do nothing except pre-publish the new KSK with ample time to give resolver administrators enough time to learn the new KSK.

Reason for the change: *Recent work on DNSSEC operations (draft-ietf-dnsop-dnssec-key-timing-00) and the revision of RFC 4641 (draft-ietf-dnsop-rfc4641bis-04) show that keeping the outgoing ZSK should be retained for a period of time. This is because validating end systems might receive cached DNS data with signatures from the old ZSK, and need to obtain the outgoing ZSK for validation. Currently, this is still a very small minority of end systems, as most are stub clients that rely on and upstream validating recursive server for DNSSEC processing.*

New References

II. DNSSEC

S. Morris, J. Ihren and J. Dickinson. "DNSSEC Key Timing Considerations" (Work in Progress) draft-ietf-dnsop-dnssec-key-timing-00 July 2010.

O. Kolkman. "DNSSEC Operational Practices, Version 2" (Work in Progress). draft-ietf-dnsop-rfc4641bis-04 August 2010.